

CSE 240 Spring 2019 Homework 6, Linked Lists (50 points)

Due Saturday, February 23, 2019 at 11:59PM, plus a 24-Hour grace period

Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures, including enumeration type, linked list, pointer operations, memory leak, parameter passing mechanisms, and recursion. By the end of the assignment, you should have understood and exercised

- Enumeration type and its operations
- Pointer operations accessing structured data
- Linked list of structures, with complex manipulations of pointer and structure
- Memory management and leak prevention
- Parameter passing and return values of different types.
- Recursive program.

Reading: Chapter 2, Textbook Section 2.5.4 on linked list and Section 2.6 on parameter passing; Section 2.10 a case study on linked list operations; and sample code in lecture slides.

Preparation: Complete the multiple choice questions in the textbook exercise section. The answer keys can be found in the course Web site. These exercises can help you prepare for your weekly quiz and the exam. You are encouraged to read the other exercise questions and make sure you understand these questions in the textbook exercise section, which can help you better understand what materials are expected to understand after the lectures and homework on each chapter.

You are expected to do the majority of the assignment outside the class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. However, **do not share your answers and code** in the course discussion board.

Programming Assignment (50 points)

1. You are given a partially completed program hw06q1.c. The structure of this homework is similar to previous homework. In this homework, you should use linked list to do the same work in the previous homework.

You should follow the instructions given in the program to complete the functions so that the program executes properly. You will be completing a program that creates a linked list of books.

It is a menu-driven program where the user is given the following options:

- a) Add a new patient to the list. When adding a new patient to the list, the user is prompted for patient's name, doctor's name, critical level of patient and room number of the patient. The patient should be added at the end of the list. If the patient already exists in the list, then you should not add to the list. The critical level is enum type.
- b) Display the list of patients. This should display each patient's details one after the other.
- c) Sort the list of patients alphabetically by patient name. The sorting should happen within the list. You should not create a new linked list of patients having sorted patients.
- d) Remove the patient from the linked list. After removing the patient, the linked list should not be broken.

Expected output of each menu option: (similar as previous homework)

add:

```
CSE240 HW6

Currently 0 patient(s) on the list.
Enter your selection:
    a: add a new patient
    d: display patient list
    r: remove a patient from the list
    s: sort patient list by name
    q: quit
a

Enter patient name: Eden Hazard
Enter doctor name: Tony Stark
Enter whether patient is 'very critical' or 'critical' or 'not critical': not critical
Please enter room number: 101

Patient successfully added to the list!

Currently 1 patient(s) on the list.
Enter your selection:
    a: add a new patient
    d: display patient list
    r: remove a patient from the list
    s: sort patient list by name
    q: quit
```

displayList:

```
Currently 3 patient(s) on the list.  
Enter your selection:  
    a: add a new patient  
    d: display patient list  
    r: remove a patient from the list  
    s: sort patient list by name  
    q: quit
```

```
d
```

```
Patient name: Eden Hazard  
Doctor name: Tony Stark  
Critical level: not critical  
Room number: 101
```

```
Patient name: Willian Borges  
Doctor name: Bruce banner  
Critical level: critical  
Room number: 155
```

```
Patient name: David Luiz  
Doctor name: Steve Rogers  
Critical level: very critical  
Room number: 290
```

deleteNode: (see displayList() photo for list before deleting node)

```
Currently 3 patient(s) on the list.
Enter your selection:
    a: add a new patient
    d: display patient list
    r: remove a patient from the list
    s: sort patient list by name
    q: quit
r

Please enter patient name: Eden Hazard

Patient successfully removed from the list.

Currently 2 patient(s) on the list.
Enter your selection:
    a: add a new patient
    d: display patient list
    r: remove a patient from the list
    s: sort patient list by name
    q: quit
d

Patient name: David Luiz
Doctor name: Steve Rogers
Critical level: very critical
Room number: 290

Patient name: Willian Borges
Doctor name: Bruce banner
Critical level: critical
Room number: 155
```

sortList: (see displayList() photo for unsorted list)

```
Currently 3 patient(s) on the list.
Enter your selection:
    a: add a new patient
    d: display patient list
    r: remove a patient from the list
    s: sort patient list by name
    q: quit
s

Patient list sorted! Use display option 'd' to view sorted list.

Currently 3 patient(s) on the list.
Enter your selection:
    a: add a new patient
    d: display patient list
    r: remove a patient from the list
    s: sort patient list by name
    q: quit
d

Patient name: David Luiz
Doctor name: Steve Rogers
Critical level: very critical
Room number: 290

Patient name: Eden Hazard
Doctor name: Tony Stark
Critical level: not critical
Room number: 101

Patient name: Willian Borges
Doctor name: Bruce banner
Critical level: critical
Room number: 155
```

Grading of Programming Assignment

The TA will grade your program following these steps:

- (1) Compile the code. If it does not compile, 20% of the points given will be deducted. For example, if there are 20 points possible, you will earn 16 points if the program fails to compile.
- (2) The graders will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementation of each function.
- (3) There will be a 10% deduction of points for not having comments in the code.

What to Submit?

You are required to submit your solution in a compressed format (.zip). Make sure your compressed file is label correctly - lastname_firstname6.zip. (All lowercase, do not put anything else in the name like "hw6".)

The compressed file MUST contain the following:

hw06q1.c (completed code)

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded. (Even if the assignment is submitted late.)

Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

Late submission deduction policy

- No penalty for late submissions that are received within 24 hours after the deadline;
- 10% grade deduction for every day it is late after the grace period;
- No late submission after Tuesday at 11:59PM.